

AD-A251 688



2

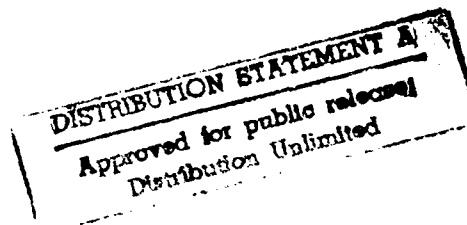
NASA Technical Memorandum 104207
AVSCOM Technical Report 92-B-006

**AN INCREMENTAL STRATEGY FOR CALCULATING
CONSISTENT DISCRETE CFD SENSITIVITY
DERIVATIVES**

Vamshi Mohan Korivi
Arthur C. Taylor III
Perry A. Newman
Gene W. Hou
Henry E. Jones



February 1992



92-15874



NASA

National Aeronautics and
Space Administration

Langley Research Center
Hampton, VA 23665-5225

US ARMY
AVIATION
SYSTEMS COMMAND
AVIATION R&T ACTIVITY



92-15874

AN INCREMENTAL STRATEGY FOR CALCULATING CONSISTENT DISCRETE CFD SENSITIVITY DERIVATIVES

Vamshi Mohan Korivi and Arthur C. Taylor III

Old Dominion University, Norfolk, VA 23529-0247

Perry A. Newman

NASA Langley Research Center, Hampton, VA 23665-5225

Gene W. Hou

Old Dominion University, Norfolk, VA 23529-0247

Henry E. Jones

USA-AVSCOM, Hampton, VA 23665-5225

ABSTRACT

In this preliminary study involving advanced CFD codes, an incremental formulation, also known as the "delta" or "correction" form, is presented for solving the very large sparse systems of linear equations which are associated with aerodynamic sensitivity analysis. For typical problems in 2D, a direct solution method can be applied to these linear equations in either the standard or the incremental form, in which case the two are equivalent. Iterative methods appear to be needed for future 3D applications, however, because direct solver methods require much more computer memory than is currently available. Iterative methods for solving these equations in the standard form result in certain difficulties, such as ill-conditioning of the coefficient matrix, which can be overcome when these equations are cast in the incremental form; these and other benefits are discussed herein. The methodology is successfully implemented and tested in 2D using an upwind, cell-centered, finite volume formulation applied to the thin-layer Navier-Stokes equations. Results are presented for two laminar sample problems: 1) transonic flow through a double-throat nozzle, and 2) flow over an isolated airfoil.

1.0 Introduction

For many complex flow fields of interest in practical engineering problems, accurate detailed analyses are now possible using supercomputers and advanced software; these codes have been developed in recent years through an intensive research effort focused in the discipline now known as Computational Fluid Dynamics (CFD). For these advanced CFD codes to become more useful as practical design tools, additional software is needed which will efficiently provide accurate aerodynamic sensitivity derivatives which are consistent with the discrete flow solutions of the particular CFD code of choice. The theme of this report is the ongoing development of a methodology for calculating these derivatives.

A sensitivity derivative is defined as the derivative of a system response of interest (e.g., the lift or drag of an airfoil) with respect to an independent design variable of interest (e.g., a parameter which controls the shape of an airfoil). In a typical design environment, a very large number of analyses are often made in determining the "best" design. An efficient method for calculating accurate sensitivity derivatives can be applied in several different ways to significantly reduce the number and/or computational cost of these multiple analyses. This could be critical for the integration of advanced CFD codes into a systematic design methodology, where the computational cost of a single flow analysis can be extremely high, particularly in 3D.

One method of a very general yet conceptually simple nature for computing aerodynamic sensitivity derivatives is the method of "brute force" finite differences. With this method, assuming forward finite difference approximations are used, the CFD flow analysis code is used to generate one converged flow solution for a slightly perturbed value of each design variable for which sensitivity derivatives are required. The principal drawback of this method is clearly that of computational cost, since the number of flow analyses required in a typical design problem can be extremely (i.e., prohibitively) large, particularly when the number of design variables is large.

As a typically less costly alternative to the finite difference approach, aerodynamic sensitivity derivatives can (in principle) be computed by direct differentiation of the governing equations which control the fluid flow. If the continuous governing equations are differentiated prior to their numerical discretization, the method is known as the "continuum" approach. In contrast, if the resulting algebraic equations which model the governing equations are differentiated following their discretization, the method is known as the "discrete" approach. In developing efficient methods for computing these sensitivity derivatives and their subsequent application to aerodynamic design problems, researchers have been and remain active; Refs. 1 through 24 are a representative (but not exhaustive) sample of articles which are germane to the present effort. Reference 8 addresses the distinction between the aforementioned "continuum" and "discrete" approaches, and Ref. 24 is a concurrent study which addresses related issues of specific interest here.

The present study represents an extension of the recent efforts of Refs. 13 through 23, where using the discrete approach, fundamental sensitivity equations are derived by direct differentiation of the system of nonlinear algebraic equations which model either the Euler or thin-layer Navier-Stokes (TLNS) equations for 2D steady flow. This differentiation results in very large systems of algebraic linear sensitivity equations which must be solved to obtain these derivatives of interest. In Refs. 13 through 23, the fundamental sensitivity equations are solved in what is

henceforth referred to herein as the "standard" (i.e., non-incremental) form. Furthermore, in these references, a direct solver method is applied to solve these equations; the single exception is Ref. 23, where a hybrid direct solver/conventional iterative approach is adopted for an isolated airfoil example problem. There are some important advantages in using a direct method when feasible; these are discussed in the references and also noted in a later section of this report. However, the most serious disadvantage of a direct solver method is the extremely large computer storage requirement, which for practical 3D problems appears to be well beyond the current capacity of modern supercomputers; this capacity can even be exceeded in 2D on very fine grids.

In an effort to circumvent the computer storage limitation for the direct methods, this preliminary study focuses on fundamental algorithm development for the efficient iterative solution of the aerodynamic sensitivity equations. That is, the principal motivation and objective is to develop a solid framework in 2D from which future extensions to 3D will be feasible. In general, one of the most serious difficulties encountered in the development and/or application of iterative techniques is that of poor overall conditioning and lack of diagonal dominance in the coefficient matrix. Unfortunately, this is a very common occurrence in the coefficient matrices of interest here; the severity varies greatly and depends on many factors. This problem can manifest itself in either poor performance or even complete failure (i.e., divergence) of an iterative algorithm.

A computationally useful property of the "incremental" form (also commonly known as the "delta" or "correction" form) can be effectively exploited to combat these problems of poor matrix conditioning. This property is that "approximations of convenience" can be introduced into the coefficient matrix of the equations, without affecting the final converged values of the sensitivity derivatives. The approximations must be "reasonable" enough that the resulting iterative strategy is convergent. In contrast, if any approximations are made to the coefficient matrix of the equations in the standard form, then the computed sensitivity derivatives cannot be consistently discrete; that is, they will not be the correct derivatives of the algebraic equations which are solved when generating the steady-state flow solution. In the implementation of the incremental formulation herein, a judiciously selected block-diagonally dominant matrix is introduced as an approximate replacement for the original ill-conditioned left-hand side coefficient matrix. The positive impact which this can have on the development of iterative techniques for the aerodynamic sensitivity equations is discussed herein, and illustrated in the example problems. Additional benefits which might be derived from this flexible nature of the "delta" formulation are also discussed.

The remainder of this report is organized as follows. The next section, presentation of theory, is further subdivided into five subsections which review and discuss: 1) the governing equations, 2) the spatial discretization and implicit formulation, 3) the fundamental sensitivity equations in standard form, 4) basic linear equation solving in incremental form, and 5) incremental solution of the equations of sensitivity analysis, where some significant implications of this formulation compared to the standard form are noted. Following the presentation of theory section, computational results are presented, illustrating application of the methodology to two laminar viscous flow example problems: 1) transonic internal flow through a double-throat nozzle and 2) external flow over an isolated airfoil. The last section is a summary where conclusions are given. In an appendix, the direction of ongoing and future work is discussed, where sample



<input checked="checked" type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
Codes	
and/or	
Special	
A-1	

results are shown from the successful application of a spatially-split approximately factored strategy for efficiently solving the sensitivity equations in incremental form.

2.0 Presentation of Theory

2.1 Governing Equations

The governing equations in this study are the 2D thin-layer Navier-Stokes (TLNS) equations; they are

$$\frac{1}{J} \frac{\partial Q}{\partial t} = R(Q) \quad (1)$$

where

$$R(Q) = -\frac{\partial \hat{F}(Q)}{\partial \xi} - \frac{\partial \hat{G}(Q)}{\partial \eta} + \frac{\partial \hat{G}_v^{tl}(Q)}{\partial \eta} \quad (2)$$

$$Q = [\rho, \rho u, \rho v, \rho e_0]^T \quad (3)$$

The vector, $R(Q)$, is known as the residual, and is clearly null for steady flow. The elements of the vector, Q , are the conserved variables, where, ρ is density, u and v are velocity components in Cartesian coordinates, and e_0 is total energy (i.e., $e_0 = e + \frac{u^2 + v^2}{2}$, where e is the specific internal energy of the fluid). The inviscid flux vectors, $\hat{F}(Q)$ and $\hat{G}(Q)$, are

$$\begin{aligned} \hat{F}(Q) &= \frac{\xi_x}{J} F(Q) + \frac{\xi_y}{J} G(Q) \\ \hat{G}(Q) &= \frac{\eta_x}{J} F(Q) + \frac{\eta_y}{J} G(Q) \end{aligned} \quad (4)$$

A transformation to generalized (ξ, η) coordinates from Cartesian (x, y) coordinates has been made in Eq. (1), where $\xi_x, \xi_y, \eta_x, \eta_y$ are "metric" terms, and J is the determinant of the Jacobian matrix of this transformation. The Cartesian flux vectors, $F(Q)$ and $G(Q)$, are

$$\begin{aligned} F(Q) &= [\rho u, \rho u^2 + p, \rho uv, (\rho e_0 + p)u]^T \\ G(Q) &= [\rho v, \rho uv, \rho v^2 + p, (\rho e_0 + p)v]^T \end{aligned} \quad (5)$$

The pressure, p , is evaluated using the ideal gas law

$$p = (\gamma - 1) \left[\rho e_0 - \rho \left(\frac{u^2 + v^2}{2} \right) \right] \quad (6)$$

and γ is the ratio of specific heats, taken to be 1.4. The thin-layer viscous terms in generalized coordinates are

$$\hat{G}_v^{tl}(Q) = \left(\frac{\mu}{Re_L} \right) [\hat{g}_{v1}, \hat{g}_{v2}, \hat{g}_{v3}, \hat{g}_{v4}]^T \quad (7)$$

where

$$\begin{aligned}
\hat{g}_{v_1} &= 0 \\
\hat{g}_{v_2} &= \alpha_1 u_\eta + \alpha_3 v_\eta \\
\hat{g}_{v_3} &= \alpha_3 u_\eta + \alpha_2 v_\eta \\
\hat{g}_{v_4} &= \frac{1}{2} \alpha_1 (u^2)_\eta + \frac{1}{2} \alpha_2 (v^2)_\eta \\
&\quad + \alpha_3 (uv)_\eta + \frac{\alpha_4}{\text{Pr}(\gamma - 1)} (a^2)_\eta \\
\alpha_1 &= \left(\frac{\eta_y^2}{J} + \frac{4}{3} \frac{\eta_x^2}{J} \right), \quad \alpha_2 = \left(\frac{\eta_x^2}{J} + \frac{4}{3} \frac{\eta_y^2}{J} \right), \\
\alpha_3 &= \left(\frac{1}{3} \frac{\eta_x \eta_y}{J} \right), \quad \alpha_4 = \frac{\eta_x^2 + \eta_y^2}{J}
\end{aligned} \tag{8}$$

The molecular viscosity is given by μ , Stokes' hypothesis for the bulk viscosity ($\lambda = -2\mu/3$) has been used, a is the speed of sound, Pr is the Prandtl number (taken to be 0.72), and Re_L is the Reynolds number. Nondimensionalization of Eq. (1) is with respect to ρ_∞ and U_∞ , the freestream density and velocity, respectively. The physical coordinates (x, y) are nondimensionalized by a reference length, L , and the viscosity is nondimensionalized by μ_∞ , the molecular viscosity of the freestream. The nondimensional molecular viscosity can be computed using Sutherland's law and a reference temperature, T_∞ , the static temperature of the freestream. For additional simplicity here, however, the molecular viscosity is taken to be constant, equal to that of the freestream.

2.2 Spatial Discretization and Implicit Formulation

Computationally, the TLNS equations are solved here in their alternative integral conservation law form using an upwind cell-centered finite volume formulation (see Refs. 25 through 31), where the residual at each cell is evaluated as a balance of inviscid and viscous fluxes across cell interfaces. Upwind evaluation of the inviscid fluxes is accomplished by upwind interpolation of the field variables, Q , from the approximate cell centers to the cell interfaces, where the flux-vector splitting procedure of van Leer (Ref. 32) is employed. In this study, third-order accuracy is used for the inviscid flux balance in the streamwise (ξ) and in the normal (η) directions. The finite volume equivalent of second-order accurate central differences is used to approximate the thin-layer viscous terms. This results in a higher-order accurate algebraic approximate representation of the residual at each cell in the domain. When assembled globally including all cells and boundary condition relationships, this can be expressed as

$$\{R(Q^*)\} = \{0\} \tag{9}$$

where $\{Q^*\}$ is called the "root" (i.e., the steady-state value of the field variables). Therefore, Eq. (9) represents a large coupled system of nonlinear algebraic equations; thus finding a steady-state solution to the TLNS equations has been replaced (approximately) by the problem of finding the root, $\{Q^*\}$, of this set of algebraic equations. In Eq. (9) and henceforth, the notation, ' $\{ \}$ ', indicates a global column vector.

The TLNS equations are discretized in time using the Euler implicit method, followed by a Taylor's series linearization of the discrete equations in time about the known time level. This results in a large system of linear algebraic equations at each time step, which is

$$\left(\left[\frac{1}{J\Delta t} \right] - \left[\frac{\partial R^n(Q)}{\partial Q} \right] \right) \{^n \Delta Q\} = \{R^n(Q)\} \quad (10)$$

$$\begin{aligned} \{Q^{n+1}\} &= \{Q^n\} + \{^n \Delta Q\} \\ n &= 1, 2, 3, \dots \end{aligned} \quad (11)$$

Equations (10) and (11) represent the fundamental implicit formulation for integrating the TLNS equations in time to steady-state. In these equations, 'n' is the time iteration index, and $\{^n \Delta Q\}$ is the incremental change in the field variables between the known (n^{th}) and the next ($n^{\text{th}}+1$) time levels. The matrix, $\left[\frac{1}{J\Delta t} \right]$, is diagonal, and contains the time term. The large Jacobian matrix, $\left[\frac{\partial R^n(Q)}{\partial Q} \right]$, is sparse and has a banded structure, with nine diagonals, the individual elements of which are 4×4 block coefficient matrices. In addition to its use in Eq. (10) above, this important Jacobian matrix plays another central role in this study, which will be shown later.

In principle, Eq. (10) can be repeatedly solved directly (using Eq. (11) to update the field variables), as the solution is advanced to steady-state; for very large time steps, the direct method represents Newton's root finding procedure for nonlinear equations. The direct method however is not necessarily the most efficient procedure with respect to overall CPU time (Ref. 33), and the large storage requirements of the method make it infeasible in 3D. Therefore, more commonly, an iterative algorithm is selected for use in the repeated solution of Eq. (10). Popular choices of these iterative algorithms include approximate factorization (AF) (Ref. 34), conventional relaxation algorithms (Refs. 29, 30), the strongly implicit procedure (SIP) (Ref. 35), and preconditioned conjugate gradient methods (Refs. 36, 37), to name a few.

It is noted that Eqs. (10) and (11) are an incremental formulation for solving the nonlinear problem of Eq. (9). If convergence is achieved, the steady-state solution, $\{Q^*\}$, only depends on what is implemented in the discrete formulation of the residual vector on the right-hand side of Eq. (10). It is also implied that this solution is independent of any approximations which are made in the coefficient matrix of Eq. (10). The final solution is also independent of the initial guess, and all transient solutions which are generated prior to convergence.

For typical advanced CFD flow codes which employ the implicit time integration formulation of Eqs. (10) and (11), the following approximations are often seen in the coefficient matrix of Eq. (10) (the list is a representative but not exhaustive one):

- 1) A first-order accurate upwind spatial discretization of the implicit terms is used, even though a higher-order accurate spatial discretization, either upwind or perhaps even central "differences" (Ref. 29), is used on the right-hand side of the equation.
- 2) A consistently linearized treatment of the boundary conditions in "delta" form is typically neglected. In particular, a fully consistent treatment of the implicit terms resulting from

the "periodic" boundary conditions of "C" and "O" meshes and also of the implicit terms across the zonal interfaces of multiblock grids is needed.

- 3) Only approximate solutions of Eq. (10) are actually generated at each time step with the use of iterative methods, in order that each time step is efficiently completed.

The preceding examples and many others not mentioned are "approximations of convenience" and are made on the left-hand side of Eq. (10) in order to influence the nature of the resulting algorithm which is to be used in finding the solution. These may be introduced for computational simplicity of implementation or overall efficiency, or both. This flexibility of the delta formulation, which allows approximations to be introduced into the left-hand side coefficient matrix without influencing the final solution, can also be exploited in the solution of the linear aerodynamic sensitivity equations, as will be seen in subsequent sections.

2.3 Fundamental Sensitivity Equations In Standard Form

Consider the vector, $\bar{\beta}$, whose elements are independent variables, typically called the design variables. Some, none, or all of these variables may be related to the geometric shape of the boundary surface of the flow problem of interest. Computationally, the geometric shape of the domain is defined by the mesh upon which calculations are made; the complete vector of (x,y) coordinates which defines the mesh is represented symbolically herein as $\{\bar{X}\}$. For a steady-state solution, the discrete residual vector given by Eq. (9) is expressed now in the following form

$$\{R(Q^*(\bar{\beta}), \bar{X}(\bar{\beta}), \bar{\beta})\} = \{0\} \quad (12)$$

where the explicit dependence of the discrete residual on the computational mesh, $\{\bar{X}\}$, as well as its explicit dependence (if any) on $\bar{\beta}$ has now been emphasized. Direct differentiation of Eq. (12) with respect to β_k , the k^{th} element of $\bar{\beta}$, yields

$$-\left[\frac{\partial R}{\partial Q}\right] \left\{ \frac{\partial Q^*}{\partial \beta_k} \right\} = \left[\frac{\partial R}{\partial \bar{X}}\right] \left\{ \frac{\partial \bar{X}}{\partial \beta_k} \right\} + \left\{ \frac{\partial R}{\partial \beta_k} \right\} \quad (13)$$

Term 1 Term 2 Term 3

Equation (13) is an exact derivative of the discrete algebraic residual vector; this procedure is known in Refs. 2 and 4 as the quasi-analytical method. The Jacobian matrix, $\left[\frac{\partial R}{\partial Q}\right]$, of Eq. (13) is identical to that found in the fundamental implicit formulation for numerical time integration (Eq. (10)) of the TLNS equations, except that is evaluated at steady-state, $\{Q^*\}$. It is thus well understood. The solution vector, $\left\{ \frac{\partial Q^*}{\partial \beta_k} \right\}$, is the sensitivity of the complete vector of field variables with respect to the k^{th} design variable. The matrix, $\left[\frac{\partial R}{\partial \bar{X}}\right]$, is the Jacobian matrix of the discrete steady-state residual vector with respect to the complete vector of (x,y) grid coordinates; it is documented in detail in Ref. 17. The vector, $\left\{ \frac{\partial \bar{X}}{\partial \beta_k} \right\}$, of Term 2, contains what is referred to here as the grid sensitivity terms; these are the sensitivity derivatives with respect to β_k of each 'x' and 'y' coordinate point of the entire computational mesh. The treatment of the terms of the grid sensitivity vector is given special consideration in Refs. 18, 23, 38, and 39. The vector,

$\left\{ \frac{\partial R}{\partial \beta_k} \right\}$, accounts for derivatives resulting from explicit dependencies (if any) of the residual vector on β_k , and additional discussion concerning this is found in Ref. 21. In the event that β_k is not a design parameter for the geometric shape, then the second term of Eq. (13) will be zero, since the vector, $\left\{ \frac{\partial \bar{X}}{\partial \beta_k} \right\}$, is then null. If β_k is a geometric shape design parameter, its effect on the residual (Eq. (12)) will usually be felt only through the grid, and the final term of Eq. (13) will generally be zero.

It is strongly emphasized that all boundary condition relationships must be treated in a fully consistent manner, and included in Eq. (13) above. Proper boundary condition treatment should be included in the Jacobian matrices, $\left[\frac{\partial R}{\partial Q} \right]$ and $\left[\frac{\partial R}{\partial \bar{X}} \right]$, as well as in the vector, $\left\{ \frac{\partial R}{\partial \beta_k} \right\}$. If accurate results are to be obtained using the present methods, it is critical that this is not neglected here as it often is in the fundamental implicit time integration formulation (i.e., Eq. (10)). Detailed documentation on the consistent treatment of the boundary conditions and its importance in these equations is found in Refs. 21, 22, and 23.

Note that Eq. (13) is a linear system of equations which in principle can be solved directly for the vector, $\left\{ \frac{\partial Q^*}{\partial \beta_k} \right\}$. Of course, the solution of Eq. (13) must be repeated for each element of $\bar{\beta}$ (i.e., for each design variable) for which sensitivity derivatives are desired. However, a single LU factorization of the coefficient matrix can be repeatedly reused for multiple solutions (i.e., for multiple design variables) in the forward and backward substitution operations. The reuse of the LU factorization can represent a substantial savings in computational work, particularly when the linear system of Eq. (13) and/or the number of design variables of interest is large.

The solution of Eq. (13) for the vector, $\left\{ \frac{\partial Q^*}{\partial \beta_k} \right\}$, is not the final goal; rather, the sensitivity derivatives of some specific system responses are sought (e.g., for an airfoil, the sensitivities of the lift, drag, and moment coefficients might be required). Consider therefore the j^{th} system response of interest, C_j , which in general can be functionally dependent on the steady-state field variables, $\{Q^*\}$, the grid, $\{\bar{X}\}$, and also explicitly on the design variables, $\bar{\beta}$; that is

$$C_j = C_j(Q^*(\bar{\beta}), \bar{X}(\bar{\beta}), \bar{\beta}) \quad (14)$$

The total rate of change of the j^{th} system response, C_j , with respect to the k^{th} design variable, β_k , is then given by

$$\frac{dC_j}{d\beta_k} = \underbrace{\left\{ \frac{\partial C_j}{\partial Q} \right\}^T \left\{ \frac{\partial Q^*}{\partial \beta_k} \right\}}_{\text{Term 1}} + \underbrace{\left\{ \frac{\partial C_j}{\partial \bar{X}} \right\}^T \left\{ \frac{\partial \bar{X}}{\partial \beta_k} \right\}}_{\text{Term 2}} + \underbrace{\frac{\partial C_j}{\partial \beta_k}}_{\text{Term 3}} \quad (15)$$

where in Eq. (15), Term(s) 2 and/or 3 could be zero, depending on the particular system response (C_j) and design variable (β_k) of concern. Solution of Eq. (13) therefore provides the vector, $\left\{ \frac{\partial Q^*}{\partial \beta_k} \right\}$, which is needed in Eq. (15). Furthermore, for geometric shape sensitivity derivatives, the grid sensitivity vector, $\left\{ \frac{\partial \bar{X}}{\partial \beta_k} \right\}$, of Eq. (13) is reused, if needed, in Eq. (15). Specific ancillary sensitivity relationships of the type given by Eq. (15) which are used in the present study for computing sensitivity derivatives of aerodynamic force coefficients are presented in Ref. 23.

On the left-hand side of Eq. (15) above, the notation for a total derivative has been used, indicating that the total rate of change of C_j with respect to β_k is included in the expression, and to distinguish it from the partial derivative term (Term 3) on the right-hand side of the expression. However, it should be understood that this derivative is still a partial derivative in the sense that C_j is in general a function of multiple independent design variables. For consistency, this notation will continue to be used throughout.

A closely related alternative procedure for computing sensitivity derivatives, known as the adjoint variable approach, is easily developed using the relationships presented thus far. This begins by combining Eqs. (13) and (15) to yield

$$\begin{aligned} \frac{dC_j}{d\beta_k} = & \left\{ \frac{\partial C_j}{\partial Q} \right\}^T \left\{ \frac{\partial Q^*}{\partial \beta_k} \right\} + \left\{ \frac{\partial C_j}{\partial \bar{X}} \right\}^T \left\{ \frac{\partial \bar{X}}{\partial \beta_k} \right\} + \frac{\partial C_j}{\partial \beta_k} \\ & + \{\lambda_j\}^T \left(\left[\frac{\partial R}{\partial Q} \right] \left\{ \frac{\partial Q^*}{\partial \beta_k} \right\} + \left[\frac{\partial R}{\partial \bar{X}} \right] \left\{ \frac{\partial \bar{X}}{\partial \beta_k} \right\} + \left\{ \frac{\partial R}{\partial \beta_k} \right\} \right) \end{aligned} \quad (16)$$

The adjoint variable vector, $\{\lambda_j\}$, is arbitrary at this point, since the inner product of $\{\lambda_j\}$ is taken with the null vector, from Eq. (13). Thus there is no net change from Eq. (15) to Eq. (16), since the entire additional term on the right-hand side of Eq. (16) is zero, for any and all $\{\lambda_j\}$. Rearranging, Eq. (16) becomes

$$\begin{aligned} \frac{dC_j}{d\beta_k} = & \left(\left\{ \frac{\partial C_j}{\partial \bar{X}} \right\}^T + \{\lambda_j\}^T \left[\frac{\partial R}{\partial \bar{X}} \right] \right) \left\{ \frac{\partial \bar{X}}{\partial \beta_k} \right\} + \frac{\partial C_j}{\partial \beta_k} + \{\lambda_j\}^T \left\{ \frac{\partial R}{\partial \beta_k} \right\} \\ & + \left(\left\{ \frac{\partial C_j}{\partial Q} \right\}^T + \{\lambda_j\}^T \left[\frac{\partial R}{\partial Q} \right] \right) \left\{ \frac{\partial Q^*}{\partial \beta_k} \right\} \end{aligned} \quad (17)$$

The necessity of evaluating the vector, $\left\{ \frac{\partial Q^*}{\partial \beta_k} \right\}$, using Eq. (13) is eliminated for all β_k by selecting the vector, $\{\lambda_j\}$, such that the coefficient of $\left\{ \frac{\partial Q^*}{\partial \beta_k} \right\}$ in Eq. (17) is null. That is, selection of $\{\lambda_j\}$ which satisfies

$$\left\{ \frac{\partial C_j}{\partial Q} \right\}^T + \{\lambda_j\}^T \left[\frac{\partial R}{\partial Q} \right] = \{0\}^T \quad (18)$$

implies

$$-\left[\frac{\partial R}{\partial Q} \right]^T \{\lambda_j\} = \left\{ \frac{\partial C_j}{\partial Q} \right\} \quad (19)$$

Therefore, following the solution of Eq. (19) for this particular choice of the adjoint variable vector, $\{\lambda_j\}$, the sensitivity derivatives of C_j with respect to all β_k are computed by

$$\frac{dC_j}{d\beta_k} = \left(\left\{ \frac{\partial C_j}{\partial \bar{X}} \right\}^T + \{\lambda_j\}^T \left[\frac{\partial R}{\partial \bar{X}} \right] \right) \left\{ \frac{\partial \bar{X}}{\partial \beta_k} \right\} + \frac{\partial C_j}{\partial \beta_k} + \{\lambda_j\}^T \left\{ \frac{\partial R}{\partial \beta_k} \right\} \quad (20)$$

Solution of the linear system of Eq. (19) for $\{\lambda_j\}$ is analogous to the solution of Eq. (13) for $\left\{\frac{\partial Q^*}{\partial \beta_k}\right\}$ in that the respective coefficient matrices are transposes of each other. A particular solution, $\{\lambda_j\}$, is valid only for a specific system response, C_j , and thus solution of Eq. (19) must be repeated for each different system response of interest. If Eq. (19) is solved directly, however, multiple solutions require only a single LU factorization of the coefficient matrix, which is repeatedly reused for an unlimited number of right-hand side vectors, $\left\{\frac{\partial C_j}{\partial Q}\right\}$ (i.e., for an unlimited number of different system responses of interest).

It is simple to verify from the preceding equations, and significant to note, that each solution, $\left\{\frac{\partial Q^*}{\partial \beta_k}\right\}$, of Eq. (13) for a particular design variable can be used for an unlimited number of different system responses. In contrast, however, each solution, $\{\lambda_j\}$, of Eq. (19) for a particular system response can be used for an unlimited number of different design variables. Therefore, in terms of computational work, if the number of system responses of interest is larger than the number of design variables, then sensitivity derivatives should be computed by solving Eq. (13). Otherwise greater computational efficiency is obtained using the adjoint variable method. Despite this difference which has been noted between these two closely related procedures, it is emphasized that the two methods are equivalent in the sense that they yield identical values for the sensitivity derivatives, if properly implemented computationally.

The significance of the well-known difference in the computational efficiency of the two methods is mitigated greatly if a direct method is used to solve the linear systems (i.e., either Eq. (13) or Eq. (19)), because the LU factorization must only be done once for multiple right-hand side vectors. However, this distinction becomes very important if an iterative strategy is used to solve these linear systems, particularly if the difference between the number of design variables and the number of system responses of interest is very large. This difference occurs, of course, because with iterative methods, the computational work required for solution of each linear system is approximately equal to the computational work required to solve the first one.

Summarizing briefly, it has been shown that calculating aerodynamic sensitivity derivatives using the discrete direct differentiation method requires the solution of large linear systems of equations of the type given by a choice of either Eq. (13) or Eq. (19). Henceforth in this report, these two systems of linear equations are known as the aerodynamic sensitivity equations in standard form. Fundamental algorithm development for the iterative solution of one of these two linear systems is easily extended and applied to the other, since as noted previously, their respective coefficient matrices are transposes of each other. In the example problems for which sensitivity derivatives are calculated in a later section, actual implementation and testing of the methods proposed herein is accomplished using Eq. (13), although the adjoint variable method, Eq. (19), could also have been used. When the linear aerodynamic sensitivity equations are solved in standard form, it should be noted that no approximations can be introduced into any of the terms, without simultaneously introducing error into the resulting sensitivity derivatives. In this form, the framework to support the development of iterative methods is thus rigid and restrictive.

As a consequence of the preceding discussion, for the higher-order accurate upwind spatial discretization which is selected herein for the flow analysis, a consistent higher-order accurate upwind spatial discretization including a fully consistent treatment of all boundary conditions is

required in the left-hand side coefficient matrix of the sensitivity equations (in standard form). Furthermore, there is no "time term" added here to enhance each element of the diagonal, as seen (in contrast) in the implicit time integration formulation of Eq. (10). Unfortunately, the resulting coefficient matrix in this case is not diagonally dominant (Ref. 29), and consequently the computational performance of traditional iterative methods for the sensitivity equations in standard form is expected to be poor, or even fail. If the present methods were applied using a popular "central difference" discretization of the inviscid terms in the flow solver, the diagonal dominance of the resulting sensitivity equations would become far worse. Therefore, it is this particular difficulty (i.e., the lack of a sufficiently strong diagonal) and how it can be overcome which is of principal concern in the development of the incremental form of the equations in the following sections.

2.4 Basic Linear Equation Solving in Incremental Form

Consider the linear system of algebraic equations in the general form

$$[A]\{Z^*\} + \{B\} = \{0\} \quad (21)$$

where $\{Z^*\}$ is the solution vector. In treating the problem of solving Eq. (21), in essence a "root finding" problem, application of Newton's method (traditionally used in root finding for nonlinear equations) to the linear problem yields the basic two-step iterative incremental formulation

$$\text{Step 1} \quad -[\tilde{A}]\{^m\Delta Z\} = [A]\{Z^m\} + \{B\} \quad (22)$$

$$\begin{aligned} \text{Step 2} \quad \{Z^{m+1}\} &= \{Z^m\} + \{^m\Delta Z\} \\ m &= 1, 2, 3, \dots \end{aligned} \quad (23)$$

where 'm' is an iteration index, and $\{^m\Delta Z\}$ is the incremental change in the solution from the known (m^{th}) to the next ($m^{\text{th}}+1$) iteration level. An initial guess, $\{Z^1\}$, is required to begin the procedure, which in the present study is taken everywhere as zero. If Newton's method is applied strictly, the coefficient matrix $[\tilde{A}]$ is equal to the matrix $[A]$, and clearly the two-step iterative strategy of Eqs. (22) and (23) for the linear problem converges on the first iteration, for any initial guess. Therefore, in this case, solution of the linear system in the standard form (Eq. (21)) and solution in the incremental form (Eqs. (22) and (23)) are equivalent.

More generally, however, the matrix $[\tilde{A}]$ is not necessarily equal to the matrix $[A]$. The matrix $[\tilde{A}]$ can be any convenient approximation of the matrix $[A]$ with the restriction that $[\tilde{A}]$ must approximate $[A]$ well enough so that the two-step iterative procedure (Eqs. (22) and (23)) converges (or, at the very least, can be forced to converge by including a strategy such as under-relaxation). Simply stated, $[\tilde{A}]$ should capture the essence of $[A]$. Furthermore, because the equations have been cast in "delta" form, the incremental method produces the unique solution of Eq. (21), $\{Z^*\}$, if convergent. In this formulation, the purpose of the left-hand side operator is to drive the right-hand side vector to zero. The final converged solution, $\{Z^*\}$, depends only on the terms on the right-hand side of Eq. (22), and thus it is emphasized here that approximations to any of these terms, including the matrix $[A]$, will produce erroneous final results.

In principle, the linear system of Eq. (22) can be solved either directly or iteratively, at each m^{th} iteration level. If a direct method is chosen, only a single LU factorization of the coefficient matrix, $[\widetilde{A}]$, is needed, where the LU factorization is then reused for an unlimited number of iterations, including when multiple solutions of Eq. (21) are sought for different values of the vector, $\{B\}$. If the coefficient matrix, $[\widetilde{A}]$, is too large, an iterative algorithm will be the only recourse because of computer storage limitations.

With the choice of an iterative algorithm, an "inner" iteration index, 'i', is established at Step 1 (Eq. (22)), and the iteration cycle over Steps 1 and 2, having index 'm', becomes the "outer" iteration loop. If the left-hand side coefficient matrix, $[\widetilde{A}]$, is diagonally dominant, then convergence of the iterative method of choice over the index, 'i', is assured for each and every linear sub-problem at Step 1. In addition, overall convergence of the procedure over the outer index, 'm', is assured if, as discussed previously, the matrix $[\widetilde{A}]$ is an adequate approximation of the matrix $[A]$, and furthermore, if each linear sub-problem at Step 1 is converged to a sufficiently close tolerance (whatever that tolerance may be).

As a simple example of the preceding discussion, if a conventional relaxation algorithm (one of many possibilities) is selected, then the matrix, $-\widetilde{[A]}$, is divided into two parts; that is

$$-\widetilde{[A]} = [M] + [N] \quad (24)$$

The iterative incremental strategy becomes

$$\begin{aligned} \text{Step 1} \quad [M]\{^{m,i}\Delta Z\} &= [A]\{Z^m\} + \{B\} - [N]\{^{m,i-1}\Delta Z\} \\ i &= 1, 2, 3, \dots, (\text{imax})^m \end{aligned} \quad (25)$$

$$\begin{aligned} \text{Step 2} \quad \{Z^{m+1}\} &= \{Z^m\} + \{^{m,(\text{imax})^m}\Delta Z\} \\ m &= 1, 2, 3, \dots \end{aligned} \quad (26)$$

where in the above, $(\text{imax})^m$ is the number of inner or sub-iterations required to converge the m^{th} linear sub-problem at Step 1 to the desired tolerance. The particular choice of the splitting in Eq. (24) of the matrix, $[\widetilde{A}]$, is made judiciously, such that Eq. (25) can be repeatedly solved very efficiently in terms of CPU time and computer storage. The most popular choices of the splitting in Eq. (24) result in either the Jacobi or the Gauss-Seidel algorithms of either the point or the line relaxation types. The use of the "delta" form line Gauss-Seidel algorithm with an "inner" and "outer" loop is investigated in Ref. 40 in the solution of the nonlinear 2D flow equations.

2.5 Incremental Solution of the Equations of Sensitivity Analysis

Application of the fundamental incremental formulation for linear equation solving, Eqs (22) and (23), to the linear system of Eq. (13) for computing aerodynamic sensitivity derivatives, gives

$$\text{Step 1} \quad - \left[\frac{\partial R}{\partial Q} \right] \left\{ {}^m \Delta \frac{\partial Q}{\partial \beta_k} \right\} = \left\{ S^m \left(\frac{\partial Q^m}{\partial \beta_k} \right) \right\} \quad (27)$$

$$\text{Step 2} \quad \left\{ \frac{\partial Q^{m+1}}{\partial \beta_k} \right\} = \left\{ \frac{\partial Q^m}{\partial \beta_k} \right\} + \left\{ {}^m \Delta \frac{\partial Q}{\partial \beta_k} \right\} \quad (28)$$

$m = 1, 2, 3, \dots$

where

$$\left\{ S^m \left(\frac{\partial Q^m}{\partial \beta_k} \right) \right\} = \left[\frac{\partial R}{\partial Q} \right] \left\{ \frac{\partial Q^m}{\partial \beta_k} \right\} + \left[\frac{\partial R}{\partial X} \right] \left\{ \frac{\partial X}{\partial \beta_k} \right\} + \left\{ \frac{\partial R}{\partial \beta_k} \right\} = \left\{ \frac{dR^m}{d\beta_k} \right\} \quad (29)$$

where the coefficient matrix $\left[\frac{\partial R}{\partial Q} \right]$ approximates the matrix $\left[\frac{\partial R}{\partial Q} \right]$, and will be discussed subsequently, in greater detail. The vector, $\left\{ S^m \left(\frac{\partial Q^m}{\partial \beta_k} \right) \right\}$, henceforth called the sensitivity residual vector, represents the total derivative of the discrete (flow analysis) residual vector, Eq. (12), with respect to β_k . From Eqs. (13) and (29), clearly the sensitivity residual vector must be driven to zero in order to find the solution, $\left\{ \frac{\partial Q^*}{\partial \beta_k} \right\}$, of Eq. (13), which is of course the objective of the incremental strategy of Eqs. (27) and (28). Approximations must not be made to any terms in the sensitivity residual vector, taking particular care that a consistent treatment of all boundary conditions is included here, if the converged solution is to yield the correct (i.e., the consistently discrete) sensitivity derivatives. The final solution at convergence depends only on the terms of this right-hand side vector.

It is proposed that a first-order accurate upwind spatial discretization of the inviscid terms is a suitable selection for use in the coefficient matrix, $\left[\frac{\partial R}{\partial Q} \right]$, of Eq. (27), as an approximation here to the higher-order accurate upwind discretization of these terms. It is believed intuitively that this approximation will be a successful choice, noting that this selection is also a common approximation of convenience which is successfully used in the coefficient matrix of the implicit time integration formulation, Eq. (10). It is most significant to note that by design, in this choice, the block-diagonal dominance is now obtained and maintained in the left-hand side coefficient matrix (Ref. 29) of Eq. (27).

In this preliminary study, the feasibility of using this first-order accurate upwind approximate treatment of the inviscid terms is investigated in the example problems. Of principal concern, of course, is whether or not this particular approximation yields a sufficiently accurate representation of these terms so that a convergent method results. However, if the proposed methodology is successful, as it is found to be in the subsequent example problems, then the door has been opened for the possible future inclusion of numerous additional "approximations of convenience" in the left-hand side coefficient matrix. Of particular interest in future studies, of course, would be some of the same previously noted approximations commonly included in the coefficient matrix of the implicit formulation for time integration (Eq. (10)) of the flow equations. In other words, typical existing CFD flow solvers (i.e., those which employ iterative delta form implicit time integration methods) might be adapted directly for use in solving the linear sensitivity equations. The feasibility of this proposal is confirmed in the appendix, where sample results are presented using the well-known spatially-split approximate factorization (AF) (Ref. 34) algorithm.

In the present preliminary examination of the proposed methodology, each linear sub-problem (i.e., Eq. (27)) is solved by direct LU factorization (followed by forward and backward substitution) using a conventional vectorized banded matrix solver (Ref. 33) which takes advantage of the fact (in terms of computational work and storage) that outside of the bandwidth, all of the elements are zero. A single complete sensitivity analysis requires a single LU factorization of the coefficient matrix, which is repeatedly reused in the forward and backward substitutions at each iteration over Eqs. (27) and (28), and for all design variables of interest. Note that the direct solution of Eq. (27) now requires only one-half of the computer storage of that which is required in the direct solution of the equations in standard form, Eq. (13), since the bandwidth of the coefficient matrix is cut in half by the use of the first-order upwind approximation. In addition, less computational work is required in the LU factorization of this coefficient matrix, and in the forward and backward substitutions (although only a single back-solving procedure is required per design variable for a direct method applied to the standard formulation).

The strategy proposed above is described as a combined iterative/direct solver method. It is felt however that the algorithm remains a direct solver method in its essential character, because despite the "factor of two" reduction in computer storage requirements, it remains infeasible for extension to practical 3D flow problems. However, the method will enable a significantly larger problem to be done in 2D. The present methodology will become purely iterative in character (and thus in principle extendable to 3D) when, as illustrated in Eqs. (25) and (26), an iterative method replaces the present direct solution of each linear sub-problem of Eq. (27). As an example given in the appendix, the AF algorithm is used to efficiently solve Eq. (27) approximately at each m^{th} iteration (without the use of sub-iterations), resulting in a convergent overall method. It is noted that convergence of iterative methods over each linear sub-problem (i.e., over each "inner loop") is assured, since $\left[\frac{\partial \widetilde{R}}{\partial Q} \right]$ is block-diagonally dominant.

Finally, it is noted here that if the adjoint variable formulation for computing the sensitivity derivatives is preferred, application of the incremental formulation to the linear system of Eq. (19) for computing the adjoint variable vector, $\{\lambda_j\}$, yields

$$\text{Step 1} \quad - \left[\frac{\partial \widetilde{R}}{\partial Q} \right]^T \{^m \Delta \lambda_j\} = \{V^m(\lambda_j^m)\} \quad (30)$$

$$\text{Step 2} \quad \left\{ \lambda_j^{m+1} \right\} = \left\{ \lambda_j^m \right\} + \{^m \Delta \lambda_j\} \quad (31)$$

$m = 1, 2, 3, \dots$

where

$$\{V^m(\lambda_j^m)\} = \left[\frac{\partial R}{\partial Q} \right]^T \{\lambda_j^m\} + \left\{ \frac{\partial C_j}{\partial Q} \right\} \quad (32)$$

The vector, $\{V^m(\lambda_j^m)\}$, known here as the adjoint variable residual vector, must be driven to zero in order to find the solution, $\{\lambda_j\}$, of Eq. (19), which is the objective of the incremental strategy of Eqs. (30) and (31).

3.0 Computational Results

Aerodynamic sensitivity derivatives computed using the incremental formulation, Eqs. (27) and (28), are presented for two laminar example problems, and are compared with the same results reported in Ref. 23 for the identical example problems. In Ref. 23, these same sensitivity derivatives were computed using direct solver based methods applied to the standard formulation of Eq. (13). It is significant to note that at the outset of this study all attempts to solve these sensitivity equations in standard form using a conventional line Gauss-Seidel iteration method (Refs. 29, 30) for these two example problems diverged, despite efforts to force convergence through the use of successive line under-relaxation. This failure is attributed to the ill-conditioning of the coefficient matrix.

3.1 Internal Flow – Double-Throat Nozzle Problem

The first example problem is that of an internal flow through a double-throat nozzle, where the flow is accelerated from a Mach number on the inflow boundary of about 0.10, to a Mach number which exceeds 2.80 at some places on the outflow boundary. The Reynolds number, RE_L , is 100, based on a reference length, L , of one-half the height of the nozzle at the smaller of the two throats. Figure 1 illustrates the geometry and computational grid which is used, and Fig. 2 depicts the Mach contours of the steady-state solution; both of these figures are taken from Ref. 23, where more complete information is given. Other studies have been conducted involving the numerical computation of flow through the geometry of this nozzle, and are documented in Refs. 41, 42, and 43.

The geometric shape is defined parametrically using analytical expressions which define the boundaries (i.e., the walls) of the nozzle. Within these analytical expressions, ten geometric shape design variables are defined, and hence these ten parameters also define the vector, $\bar{\beta}$. These ten design variables, β_1 through β_{10} , the analytical functions which define this geometric shape, and also the treatment of the grid sensitivity vectors, $\left\{ \frac{\partial \bar{X}}{\partial \beta_1} \right\}$ through $\left\{ \frac{\partial \bar{X}}{\partial \beta_{10}} \right\}$, are fully explained in Refs. 23 and 43.

In Ref. 23, the sensitivity derivatives were computed (with respect to β_1 through β_{10}) of the force coefficients, C_x and C_y ; these force coefficients are the integrated (over the lower surface) pressure and skin frictions coefficients, which have been resolved in the 'x' and 'y' directions, respectively. In this earlier study, these terms were calculated by direct solution of the aerodynamic sensitivity equations in standard form (Eq. (13)), where a single LU factorization was used in the back-solving operations for all ten design variables. Additionally in the previous work, the accuracy of the calculations was successfully validated using the method of "brute force" finite differences, and thus this consistency check is not repeated here.

In Table 1, the sensitivity derivatives of C_x and C_y with respect to the first geometric shape design variable, β_1 , are presented. The computed values of $\left\{ \frac{dC_x}{d\beta_1} \right\}$ and $\left\{ \frac{dC_y}{d\beta_1} \right\}$ are presented here from the solution of the aerodynamic sensitivity equations in incremental form, where results are given for successively larger reductions in the average global error. Specifically, the sensitivity derivatives computed using the incremental method are given for a zero, one, two, three, and four orders-of-magnitude (OM) reduction in the L_2 norm of the sensitivity residual

vector, Eq. (29), which is also the right-hand-side of Eq. (27). In addition, the number of iterations (over the two-step scheme, Eqs. (27) and (28)) which were required to achieve each of these successive levels of convergence is also included in the table. In the last row of the table, the results which were obtained by direct solution of the standard form of the equations, taken from Ref. 23, also are given. Tables 2 through 10 show results similar to those shown in Table 1, except that sensitivity derivatives of C_x and C_y with respect to β_2 through β_{10} , respectively, are presented.

For this first example problem, from the results presented in these tables, it is verified that the diagonally dominant first-order accurate upwind spatial discretization of the inviscid terms in the matrix, $\left[\frac{\partial R}{\partial Q}\right]$, of Eq. (27) is a sufficiently accurate approximation of the matrix, $\left[\frac{\partial R}{\partial Q}\right]$, that the iterative incremental formulation for solving these equations is convergent. It is noted that these results were obtained without the use of under-relaxation or any scheme to "force" the method to converge. The solutions appear to be fairly well converged after only a two OM reduction of the error, and the first four digits (at least) of these sensitivity derivatives do not change as the error is reduced from three to four OM. Most importantly, the expected result is noted (as a consistency check), that the "tightly" converged results obtained using the incremental formulation agree with the results of Ref. 23 which were obtained using the standard formulation.

Strategy Used	Error Reduction	Number of Iterations	$\frac{dC_x}{d\beta_1}$	$\frac{dC_y}{d\beta_1}$
Incremental Method, Eqs. (27), (28), (29)	0 OM*	1	-3.877 E+01	-3.211 E+02
	1 OM	13	-4.934 E+01	-3.024 E+02
	2 OM	20	-4.925 E+01	-3.024 E+02
	3 OM	27	-4.925 E+01	-3.024 E+02
	4 OM	33	-4.925 E+01	-3.024 E+02
Standard Form, Direct Solution of Eq. (13)	N/A	N/A	-4.925 E+01	-3.024 E+02

Table 1 – Comparison of Sensitivity Derivatives, Incremental and Standard Methods, First Design Variable, β_1

*OM Refers to the number of Orders-of-Magnitude reduction in the average global error.

Strategy Used	Error Reduction	Number of Iterations	$\frac{dC_x}{d\beta_2}$	$\frac{dC_y}{d\beta_2}$
Incremental Method, Eqs. (27), (28), (29)	0 OM	1	-4.644 E+02	+2.152 E+01
	1 OM	8	-4.614 E+02	+1.733 E+01
	2 OM	15	-4.614 E+02	+1.742 E+01
	3 OM	22	-4.614 E+02	+1.741 E+01
	4 OM	33	-4.614 E+02	+1.741 E+01
Standard Form, Direct Solution of Eq. (13)	N/A	N/A	-4.614 E+02	+1.741 E+01

Table 2 – Comparison of Sensitivity Derivatives, Incremental and Standard Methods, Second Design Variable, β_2

Strategy Used	Error Reduction	Number of Iterations	$\frac{dC_x}{d\beta_3}$	$\frac{dC_y}{d\beta_3}$
Incremental Method, Eqs. (27), (28), (29)	0 OM	1	+2.343E +02	-3.655 E+01
	1 OM	11	+2.284 E+02	-2.616 E+01
	2 OM	18	+2.284 E+02	-2.625 E+01
	3 OM	24	+2.284 E+02	-2.625 E+0
	4 OM	31	+2.284 E+02	-2.625 E+01
Standard Form, Direct Solution of Eq. (13)	N/A	N/A	+2.284 E+02	-2.625 E+01

Table 3 – Comparison of Sensitivity Derivatives, Incremental and Standard Methods, Third Design Variable, β_3

Strategy Used	Error Reduction	Number of Iterations	$\frac{dC_x}{d\beta_4}$	$\frac{dC_y}{d\beta_4}$
Incremental Method, Eqs. (27), (28), (29)	0 OM	1	-2.694 E+04	+2.213 E+03
	1 OM	10	-2.665 E+04	+1.659 E+03
	2 OM	17	-2.665 E+04	+1.665 E+03
	3 OM	23	-2.665 E+04	+1.664 E+03
	4 OM	31	-2.665 E+04	+1.664 E+03
Standard Form, Direct Solution of Eq. (13)	N/A	N/A	-2.665 E+04	+1.664 E+03

Table 4 – Comparison of Sensitivity Derivatives, Incremental and Standard Methods, Fourth Design Variable, β_4

Strategy Used	Error Reduction	Number of Iterations	$\frac{dC_x}{d\beta_5}$	$\frac{dC_y}{d\beta_5}$
Incremental Method, Eqs. (27), (28), (29)	0 OM	1	-8.334 E+01	+7.905 E-01
	1 OM	3	-8.326 E+01	+4.500 E-01
	2 OM	6	-8.327 E+01	+4.344 E-01
	3 OM	26	-8.327 E+01	+4.370 E-01
	4 OM	46	-8.327 E+01	+4.365 E-01
Standard Form, Direct Solution of Eq. (13)	N/A	N/A	-8.327 E+01	+4.365 E-01

Table 5 – Comparison of Sensitivity Derivatives, Incremental and Standard Methods, Fifth Design Variable, β_5

Strategy Used	Error Reduction	Number of Iterations	$\frac{dC_x}{d\beta_6}$	$\frac{dC_y}{d\beta_6}$
Incremental Method, Eqs. (27), (28), (29)	0 OM	1	+8.628 E-01	+1.421 E+02
	1 OM	9	-3.657 E-02	+1.429 E+02
	2 OM	15	-1.667 E-02	+1.428 E+02
	3 OM	22	-1.368 E-02	+1.428 E+02
	4 OM	31	-1.370 E-02	+1.428 E+02
Standard Form, Direct Solution of Eq. (13)	N/A	N/A	-1.370 E-02	+1.428 E+02

Table 6 – Comparison of Sensitivity Derivatives, Incremental and Standard Methods, Sixth Design Variable, β_6

Strategy Used	Error Reduction	Number of Iterations	$\frac{dC_x}{d\beta_7}$	$\frac{dC_y}{d\beta_7}$
Incremental Method, Eqs. (27), (28), (29)	0 OM	1	+2.120 E+00	-5.216 E-01
	1 OM	12	+1.444 E+00	-5.873 E+00
	2 OM	18	+1.414 E+00	-5.877 E+00
	3 OM	25	+1.415 E+00	-5.879 E+00
	4 OM	31	+1.415 E+00	-5.879 E+00
Standard Form, Direct Solution of Eq. (13)	N/A	N/A	+1.415 E+00	-5.879 E+00

Table 7 – Comparison of Sensitivity Derivatives, Incremental and Standard Methods, Seventh Design Variable, β_7

Strategy Used	Error Reduction	Number of Iterations	$\frac{dC_x}{d\beta_8}$	$\frac{dC_y}{d\beta_8}$
Incremental Method, Eqs. (27), (28), (29)	0 OM	1	-3.415 E-01	+2.353 E+02
	1 OM	15	+6.281 E+00	+2.331 E+02
	2 OM	21	+6.236 E+00	+2.331 E+02
	3 OM	28	+6.236 E+00	+2.331 E+02
	4 OM	33	+6.236 E+00	+2.331 E+02
Standard Form, Direct Solution of Eq. (13)	N/A	N/A	+6.236 E+00	+2.331 E+02

Table 8 – Comparison of Sensitivity Derivatives, Incremental and Standard Methods, Eighth Design Variable, β_8

Strategy Used	Error Reduction	Number of Iterations	$\frac{dC_x}{d\beta_9}$	$\frac{dC_y}{d\beta_9}$
Incremental Method, Eqs. (27), (28), (29)	0 OM	1	-1.366 E+00	-2.382 E+01
	1 OM	12	-2.153 E+00	-2.082 E+01
	2 OM	18	-2.107 E+00	-2.082 E+01
	3 OM	25	-2.107 E+00	-2.081 E+01
	4 OM	31	-2.107 E+00	-2.081 E+01
Standard Form, Direct Solution of Eq. (13)	N/A	N/A	-2.107 E+00	-2.081 E+01

Table 9 – Comparison of Sensitivity Derivatives, Incremental and Standard Methods, Ninth Design Variable, β_9

Strategy Used	Error Reduction	Number of Iterations	$\frac{dC_x}{d\beta_{10}}$	$\frac{dC_y}{d\beta_{10}}$
Incremental Method, Eqs. (27), (28), (29)	0 OM	1	+9.750 E-02	+1.144 E+01
	1 OM	7	+3.988 E-01	+1.157 E+01
	2 OM	13	+3.903 E-01	+1.158 E+01
	3 OM	20	+3.886 E-01	+1.158 E+01
	4 OM	26	+3.886 E-01	+1.158 E+01
Standard Form, Direct Solution of Eq. (13)	N/A	N/A	+3.886 E-01	+1.158 E+01

Table 10 – Comparison of Sensitivity Derivatives, Incremental and Standard Methods, Tenth Design Variable, β_{10}

Table 11 shows a comparison of total CPU times, where naturally the computational cost of the incremental method depends heavily on the “strictness” of the desired convergence tolerance. For only a two OM error reduction, the computational cost of the incremental and the standard formulations are approximately equal. However, a tightly converged (four OM error reduction) solution results in a factor of almost two greater computational cost for the incremental method in the present example problem.

Strategy Used	Error Reduction	CPU Time (Seconds)*
Incremental Method, Eqs. (27), (28), (29)	0 OM	27
	1 OM	51
	2 OM	68
	3 OM	90
	4 OM	113
Standard Form, Direct Solution of Eq. (13)	N/A	66

Table 11 – Comparison of Total CPU Times, Incremental and Standard Methods
*All Calculations Performed on a Cray-2 Computer.

It is noted that the sensitivity derivatives presented in the first row of Tables 1 through 10 (i.e., for a zero OM error reduction, which is one iteration of the incremental method) are exactly the values which would be computed by direct solution of the standard formulation, if the left-hand side coefficient matrix, $\left[\frac{\partial R}{\partial Q}\right]$, of Eq. (13) were approximated using the matrix, $\left[\widetilde{\frac{\partial R}{\partial Q}}\right]$. By comparison of these calculations in the first row with those in the last row of the tables (i.e., the actual results of the standard formulation), the significant error is seen which would be generated in the sensitivity derivatives if approximations of convenience such as this were introduced into the standard formulation of the equations.

3.2 External Flow – NACA 4-Digit Airfoil Problem

The second problem considered here is that of external flow over an isolated airfoil, and is identical to the second example problem of Ref. 23. There pertinent details are found, including the grid and boundary conditions used, as well as an explanation of the special treatment for the grid sensitivity terms. The numerical solution of this laminar flow problem is for a freestream Mach number, $M_\infty = 0.70$, Reynolds number, $Re_L = 5000$, and angle of attack $\alpha = 0.0^\circ$. The airfoil shape is the NACA 2412, where the profile is defined by polynomial expressions in terms of three parameters, which are maximum thickness, $T=0.12$, maximum camber, $C=0.02$, and location of maximum camber, $L=0.40$. These three parameters are defined here to be the design variables, and hence define the elements of the vector, β .

In Ref. 23, sensitivity derivatives were computed (with respect to T , C , and L) for the lift (C_L) and drag (C_D) coefficients. These terms were calculated in this earlier work using a hybrid direct solver/conventional iterative approach in the solution of the sensitivity equations in standard form (Eq. (13)). That is, a single direct LU factorization was applied to the central bandwidth of the coefficient matrix; the relatively small number of implicit terms which fall outside this main bandwidth (some at extreme distances because of the "periodic" boundary conditions at the "wake-cut" of the C-mesh) were treated "explicitly," i.e., on the right-hand side of the equations. Thus a conventional Richardson iterative cycle was established to account for the periodic boundary conditions. However, despite the relatively small number of terms which were treated explicitly, it was reported that because of the required use of the poorly conditioned higher-order accurate coefficient matrix, the iterative strategy was at first divergent, and the use of under-relaxation was necessary to force the procedure to converge. As in the first example problem, the accuracy of the final results was successfully verified in this earlier work by finite differences, and thus this consistency check is omitted here.

In the present application of the incremental strategy to this identical airfoil problem, the elements falling outside the central bandwidth of the left-hand coefficient side matrix, $\left[\widetilde{\frac{\partial R}{\partial Q}}\right]$, were simply neglected entirely. This of course constitutes the inclusion of a second approximation of convenience in this matrix, in addition to the first-order accurate upwind treatment of the inviscid terms. The analogous (but not identical) terms resulting from the C-mesh type periodic boundary conditions in the matrix, $\left[\frac{\partial R}{\partial Q}\right]$, are not and must not be neglected on the right-hand side of Eq. (27), if the final sensitivity derivatives are to be correct. However, the treatment of these periodic terms is explicit and straightforward since they are on the right-hand side of

the equations. The resulting incremental strategy is again found to be convergent in the present example problem, without the need for under-relaxation or any scheme to force the convergence. As in the first example problem, the method is implemented by a single direct LU factorization of the approximate coefficient matrix, $\left[\frac{\partial R}{\partial Q} \right]$, which is repeatedly reused in all subsequent back-solving operations, for all iterations and design variables.

Table 12 shows the computed sensitivity derivatives of C_L and C_D with respect to $\beta_1 = T$, for successively larger reductions in the error, where the results of the present incremental formulation are compared with the results for the standard formulation, taken from Ref. 23. Tables 13 and 14 provide similar results except that derivatives with respect to $\beta_2 = C$ and $\beta_3 = L$, respectively, are computed. Note that in these tables, the convergence of each method is fairly good after a two OM reduction in the error, and excellent after three or four OM. In addition, the converged results of the standard and incremental formulations are seen to consistently agree with one another, as expected. Table 15 presents the number of iterations required to achieve each level of error reduction, for each design variable, where the incremental and standard formulations are compared. Finally, Table 16 compares the total CPU times which were required in the calculations using the incremental and standard forms. For the present problem, the incremental method is seen to be more efficient.

Error Reduction	Lift Sensitivity $\frac{dC_L}{d\beta_1} = \frac{dC_L}{dT}$		Drag Sensitivity $\frac{dC_D}{d\beta_1} = \frac{dC_D}{dT}$	
	Standard	Incremental	Standard	Incremental
0 OM	-9.334 E-01	-2.467 E-01	+4.723 E-01	+1.226 E+00
1 OM	-2.589 E+00	-2.939 E+00	+4.267 E-01	+4.353 E-01
2 OM	-3.117 E+00	-3.126 E+00	+3.972 E-01	+3.938 E-01
3 OM	-3.126 E+00	-3.126 E+00	+3.939 E-01	+3.938 E-01
4 OM	-3.126 E+00	-3.126 E+00	+3.938 E-01	+3.938 E-01

Table 12 – Comparison of Sensitivity Derivatives, Incremental and Standard Methods, $\beta_1 = T$ (Maximum Thickness)

Error Reduction	Lift Sensitivity $\frac{dC_L}{d\beta_2} = \frac{dC_L}{dC}$		Drag Sensitivity $\frac{dC_D}{d\beta_2} = \frac{dC_D}{dC}$	
	Standard	Incremental	Standard	Incremental
0 OM	+5.206 E+00	+4.706 E+00	+3.429 E-01	+6.778 E-01
1 OM	+4.175 E+00	+2.973 E+00	+3.780 E-01	+3.785 E-01
2 OM	+3.988 E+00	+3.976 E+00	+3.663 E-01	+3.640 E-01
3 OM	+3.968 E+00	+3.968 E+00	+3.603 E-01	+3.603 E-01
4 OM	+3.968 E+00	+3.968 E+00	+3.603 E-01	+3.603 E-01

Table 13 – Comparison of Sensitivity Derivatives, Incremental and Standard Methods, $\beta_2 = C$ (Maximum Camber)

Error Reduction	Lift Sensitivity $\frac{dC_L}{d\beta_3} = \frac{dC_L}{dL}$		Drag Sensitivity $\frac{dC_D}{d\beta_3} = \frac{dC_D}{dL}$	
	Standard	Incremental	Standard	Incremental
0 OM	-4.293 E-02	-8.869 E-02	-3.899 E-03	-5.195 E-03
1 OM	-1.466 E-02	-1.745 E-01	-3.422 E-03	-3.017 E-03
2 OM	-1.869 E-02	-1.833 E-02	-3.334 E-03	-3.320 E-03
3 OM	-1.819 E-02	-1.816 E-02	-3.304 E-03	-3.295 E-03
4 OM	-1.816 E-02	-1.816 E-02	-3.290 E-03	-3.290 E-03

Table 14 – Comparison of Sensitivity Derivatives, Incremental and Standard Methods, $\beta_3 = L$ (Location Of Maximum Camber)

Error Reduction	Number of Iterations Standard			Number of Iterations Incremental		
	T	C	L	T	C	L
0 OM	1	1	1	1	1	1
1 OM	13	14	5	12	14	9
2 OM	64	39	24	136	45	32
3 OM	219	188	49	239	203	91
4 OM	300	276	195	297	269	225

Table 15 – Number of Iterations Required, Incremental and Standard Methods

Error Reduction	Total CPU Time (Seconds)	
	Standard	Incremental
0 OM	27	10
1 OM	33	15
2 OM	54	41
3 OM	124	89
4 OM	191	127

Table 16 – A Comparison of Total CPU Times, Incremental and Standard Methods

4.0 Summary and Conclusions

It has been shown herein that for the future development and application of efficient iterative methods for solving the aerodynamic sensitivity equations, there are significant advantages which can be exploited within the incremental formulation which are not seen in the standard form of these equations. These benefits are derived from the flexibility of the "delta" formulation, which allows any convenient approximation to be introduced into the left-hand side coefficient matrix (which operates on the "delta terms") without affecting the final computed values of the sensitivity derivatives, provided the resulting sequence of successive iterations which are generated converges. Future efforts in algorithm development can now be directed at solving the sensitivity equations in delta form using conventional iterative strategies which are commonly employed in solving the nonlinear flow equations. The goal is to adapt existing CFD flow solvers in 2D and 3D with few or no changes to also solve the equations of aerodynamic sensitivity analysis. In this regard, preliminary results obtained to date are encouraging; in the appendix the feasibility of this proposal is confirmed in the example problems using a fully iterative solution process.

5.0 References

1. Yates, E.C., Jr., and Desmarais, R., "Boundary Integral Method for Calculating Aerodynamic Sensitivities with Illustration for Lifting Surface Theory," in Proceedings of the International Symposium of Boundary Element Methods (IBEM 89), published by Springer-Verlag, Oct. 2-4, 1989, East Hartford, Conn.
2. Elbanna, H.M., and Carlson, L.A., "Determination of Aerodynamic Sensitivity Coefficients in the Transonic and Supersonic Regimes," Journal of Aircraft, Vol. 27, No. 6, June 1990, pp. 507-518 (also AIAA Paper 89-0532).
3. Yates, E.C., Jr., "Aerodynamic Sensitivities from Subsonic, Sonic, and Supersonic Unsteady, Nonplanar Lifting-Surface Theory," NASA TM-100502, September, 1987.
4. Sobieszcanski-Sobieski, J., "The Case For Aerodynamic Sensitivity Analysis". In Sensitivity Analysis in Engineering, NASA CP-2457, 1987.
5. Bristow, D.R., and Hawk, J.D., "Subsonic Panel Method For The Efficient Analysis of Multiple Geometry Perturbations," NASA CP-3528, 1982.
6. Jameson, A., "Aerodynamic Design Via Control Theory," NASA CR-181749, ICASE Report No. 88-64, November, 1988.
7. Frank P.D., and Shubin, G.R., "A Comparison of Optimization-Based Approaches for a Model Computational Aerodynamics Design Problem," Applied Mathematics Technical Report. ECA-TR-136-R1, Boeing Computer Services, Seattle, Washington, October, 1990, also to appear, Journal of Computational Physics.
8. Shubin, G.R., and Frank, P.D. "A Comparison of the Implicit Gradient Approach and the Variational Approach to Aerodynamic Design Optimization," Applied Mathematics and Statistics Technical Report, AMS-TR-163, Boeing Computer Services, Seattle, Washington, April, 1991.
9. Shubin, G.R., "Obtaining 'Cheap' Optimization Gradients from Computational Aerodynamics Codes", Applied Mathematics and Statistics Technical Report, AMS-TR-164, Boeing Computer Services, Seattle, Washington, June 1991.
10. Sorensen, T.M., "Viscous Airfoil Optimization Using Conformal Mapping Coefficients as Design Variables," M.S. Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, June, 1991.
11. Dukravich, G. S., "Aerodynamic Shape Design and Optimization," AIAA Paper 91-0476.
12. Ta'asan, S., Kuruwila, G., and Salas, M. D., "Aerodynamic Design and Optimization in One Shot," AIAA Paper 92-0025.
13. Baysal, O. and Eleshaky, M.E., "Aerodynamic Sensitivity Analysis Methods for the Compressible Euler Equations," Recent Advances in Computational Fluid Dynamics, (ed. O. Baysal), ASME-FED Vol. 103, 11th Winter Annual Meeting, November, 1990, pp. 191-202.

14. Baysal, O., and Eleshaky, M.E., "Aerodynamic Design Optimization Using Sensitivity Analysis and Computational Fluid Dynamics," AIAA Paper 91-0471.
15. Baysal, O., Eleshaky, M.E., and Burgreen, G.W., "Aerodynamic Shape Optimization Using Sensitivity Analysis on Third-Order Euler Equations," Proceedings of the AIAA 10th Computational Fluid Dynamics Conference, June 24-26, 1991, Honolulu, Hawaii, AIAA Paper 91-1577.
16. Eleshaky, M.E., and Baysal, O., "Airfoil Shape Optimization Using Sensitivity Analysis on Viscous Flow Equations," Symposium On Multidisciplinary Applications Of Computational Fluid Dynamics, (ed. O. Baysal), ASME 12th Winter Annual Meeting, December 1-6, 1991, Atlanta, Georgia.
17. Taylor, A.C. III, Korivi, V.M., and Hou, G.W., "Sensitivity Analysis Applied to The Euler Equations: A Feasibility Study with Emphasis on Variation of Geometric Shape," AIAA Paper 91-0173 (also to appear, AIAA Journal, September, 1992, as "A Taylor's Series Approximation Of Geometric Shape Variation For The Euler Equations").
18. Taylor, A.C. III, Hou, G.W., and Korivi, V.M., "A Methodology for Determining Aerodynamic Sensitivity Derivatives With Respect to Variation of Geometric Shape," Proceedings of the AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics, and Materials Conference, April 8-10, Baltimore, MD, AIAA Paper 91-1101 (also to appear, AIAA Journal, as "A Methodology For Calculating Aerodynamic Sensitivity Derivatives").
19. Korivi, V.M., "Sensitivity Analysis Applied to the Euler Equations," M.S. Thesis, Old Dominion University, Norfolk, VA, August, 1991.
20. Taylor, A.C., III, Korivi, V.M., and Hou, G.W. "Approximate Analysis and Sensitivity Analysis Methods For Viscous Flow Involving Variation of Geometric Shape," Proceedings of the AIAA 10th Computational Fluid Dynamics Conference, June 24-26, 1991, Honolulu, Hawaii, AIAA Paper 91-1569.
21. Taylor, A.C. III, Hou, G.W., and Korivi, V.M., "An Efficient Method For Estimating Steady-State Numerical Solutions to the Euler Equations", AIAA Paper 91-1680.
22. Hou, G.W., Taylor, A.C. III, and Korivi, V.M., "Discrete Shape Sensitivity Equations For Aerodynamic Problems," AIAA Paper 91-2259.
23. Taylor, A.C. III, Hou, G.W., and Korivi, V.M., "Sensitivity Analysis, Approximate Analysis, and Design Optimization For Internal and External Viscous Flows," AIAA Paper 91-3083.
24. Newman, P.A., Hou, G.J.-W., Jones, H.E., Taylor, A.C. III, and Korivi, V.M., "Observations On Computational Methodologies For Use In Large-Scale Gradient-Based Multidisciplinary Design Incorporating Advanced CFD Codes", NASA TM-104206, February, 1992.
25. Walters, R.W., and Thomas, J.L., "Advances in Upwind Relaxation Methods," in State of the Art Surveys of Computational Mechanics, ed. A.K. Noor, ASME Publication, 1988.
26. Thomas, J.L., Van Leer, B., and Walters, R.W., "Implicit Flux-Split Schemes for the Euler Equations," AIAA Journal, Vol. 28, No. 6, June 1990, pp. 973-974 (also AIAA Paper 85-1680).

27. Walters, R.W., and Dwoyer, D.L., "An Efficient Iteration Strategy for the Euler Equations," AIAA Paper 85-1529.
28. Newsome, R.W., Walters, R.W., and Thomas, J.L., "An Efficient Iteration Strategy for Upwind/Relaxation Solutions to the Thin-Layer Navier-Stokes Equations," AIAA Journal, Vol. 27, No. 9, September, 1989, pp. 1165-1166 (also AIAA Paper 87-1113).
29. Thomas, J.L., and Walters, R.W., "Upwind Relaxation Algorithms for the Navier-Stokes Equations," AIAA Journal, Vol. 25, No. 4, April 1987, pp. 527-534.
30. Napolitano, M., and Walters, R.W., "An Incremental Block-Line-Gauss-Seidel Method for the Navier-Stokes Equations," AIAA Journal, Vol. 24, No. 5, May 1986, pp. 770-776.
31. Walters, R.W., and Dwoyer, D.I., "Efficient Solutions to the Euler Equations for Supersonic Flow with Embedded Subsonic Regions," NASA TP 2523, January 1987.
32. Van Leer, B., "Flux-Vector Splitting for the Euler Equations," ICASE Report 82-30, September 1982 (also Lecture Notes in Physics, Vol. 170, 1982, pp. 507-512).
33. Riggins, D.W., Walters, R.W., and Pelletier, D., "The Use of Direct Solvers for Compressible Flow Computations," AIAA Paper 88-0229.
34. Beam, R.M. and Warming, R.F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," AIAA Journal, Vol. 16, April 1978, pp. 393-402.
35. Walters, R.W., Dwoyer, D.L., and Hassan, H.A., "A Strongly Implicit Procedure for the Compressible Navier Stokes Equations," AIAA Journal, Vol. 24, No. 1, January 1986, pp. 6-12.
36. Ajmani, K., Ng, W.F., and Liou, M.S., "Generalized Conjugate-Gradient Methods for the Navier-Stokes Equations", Proceedings of the AIAA 10th Computational Fluid Dynamics Conference, June 24-26, 1991 Honolulu, Hawaii, AIAA Paper 91-1556.
37. Venkatakrishnan, V., "Preconditioned Conjugate Gradient Method For The Compressible Navier-Stokes Equations", AIAA Journal, Vol. 29, No. 7, July, 1991, pp. 1092-1100, also AIAA Paper 90-0586.
38. Smith, R.E., Jr., and Sadreghighi, I., "Grid Sensitivity in Airplane Design," in Proceedings of the 4th International Symposium of Computational Fluid Dynamics, September 9-12 1991, University of California-Davis, pp. 1071-1077.
39. Sadreghighi, I., Smith, R.E., Jr., and Tiwari, S.N., "An Analytical Approach To Grid Sensitivity Analysis," AIAA Paper 92-0660.
40. Taylor, A.C., III, Ng, W.F., and Walters, R.W., "Upwind Relaxation Algorithms for the Navier-Stokes Equations Using Inner Iterations," Proceedings of the AIAA 9th Computational Fluid Dynamics Conference, June 13-15, 1989, Buffalo, New York, AIAA Paper 89-1954 (also to appear, Journal of Computational Physics).
41. Viviand, H., "Comparison of Numerical Solutions to Internal Flow in a Double-Throat Nozzle," GAMM Committee for Numerical Methods in Fluid Mechanics, France, December, 1985.

42. Thomas, J.L., Walters, R.W., Van Leer, B., and Rumsey, C.L., "An Implicit Flux-Split Algorithm for the Navier-Stokes Equations," GAMM Committee for Numerical Methods in Fluid Mechanics, France, December, 1985.
43. Bristeau, M.O., Glowinski, R., Periaux, J., and Viviand, Henri (Eds.), "Numerical Simulation of Compressible Navier-Stokes Flows," Notes on Numerical Fluid Mechanics, (Proceedings of a GAMM Workshop, Numerical Methods in Fluid Mechanics), Vol. 18, pp. 5-10, published by Friedr. Vieweg, & Sohn, Braunschweig, Germany, 1987.

6.0 Appendix – Future Work, An Approximately Factored Method

Having developed and successfully demonstrated an incremental formulation which is flexible in character for solving the sensitivity equations, future work in algorithm development for the iterative solution of these equations will seek to adapt iterative strategies which are commonly used in the implicit time integration of the flow equations. To this end, a false time term, which is the diagonal matrix, $\left[\frac{1}{J\Delta t}\right]$, is added to the left-hand side coefficient matrix, $\left[\frac{\partial R}{\partial Q}\right]$, of Eq. (27). This "time" term diagonal matrix is of course found in the implicit time integration formulation of Eq. (10).

The addition of this false time term to each element on the diagonal of the coefficient matrix in Eq. (27) is equivalent to the use of under-relaxation in the two-step incremental formulation of Eqs. (27) and (28). Then, for small to moderate time steps, the resulting linear system of Eq. (27) may be very efficiently solved (approximately) at each iteration (i.e., at each false time step) using the spatially-split approximate factorization algorithm (AF) of Ref. 34. This basic algorithm, which has many variations, is well known as a common strategy found in 2D and 3D CFD flow codes for the efficient approximate solution of Eq. (10) at each time step.

With the introduction of the false time term to the elements on the diagonal, and the resulting factorization error which is associated with the AF algorithm at each iteration (all in addition to the error of the approximate first-order accurate upwind treatment of the inviscid terms), it was not known a priori whether the resulting approximate coefficient matrix operator on the left-hand side of Eq. (27) would be a convergent method for solving these equations. However, the proposed AF strategy has been found to be convergent in application to the two previously explained example problems of this study. That is, the algorithm was successfully used to produce a four OM reduction in the average error (as defined previously) for the double-throat nozzle problem and the airfoil problem.

Using a constant Courant number of 10 for each cell in the computational grid (i.e., using local false "time-stepping"), for the double-throat nozzle example, Table 17 shows the computed sensitivity derivatives of C_x and C_y (along the lower wall) with respect to β_1 through β_{10} , following the four OM reduction in the average error, where the number of iterations required by this algorithm to achieve this level of convergence is reported for each design variable. As expected, these results shown here agree very well with those reported earlier for this example problem, except some of the very small sensitivity derivatives show minor discrepancies which prove to disappear when the AF method is used to reduce the error to a stricter tolerance than

the four OM shown here. Table 18 presents a comparison of the total CPU time required in this example using the AF method compared to the CPU times shown earlier for the other methods.

Design Variable	Number of Iterations	Sensitivity of C_x	Sensitivity of C_y
β_1	335	-4.926 E+01	-3.024 E+02
β_2	277	-4.614 E+02	+1.741 E+01
β_3	242	+2.284 E+02	-2.625 E+01
β_4	276	-2.665 E+04	+1.664 E+03
β_5	259	-8.327 E+01	+4.370 E-01
β_6	278	-1.778 E-02	+1.428 E+02
β_7	225	+1.414 E+00	-5.881 E+00
β_8	317	+6.233 E+00	+2.331 E+02
β_9	280	-2.109 E+00	-2.081 E+01
β_{10}	243	+3.881 E-01	+1.158 E+01

Table 17 - Double-Throat Nozzle Problem, Approximately Factored (AF) Incremental Method, Four OM Error Reduction

Strategy Used	Total CPU Time (Seconds)
Incremental, AF Solver, (4 OM)	144
Incremental, Direct Solver (4 OM)	113
Standard Form, Direct Solution	66

Table 18 - Double-Throat Nozzle Problem, Comparison of Total CPU Times

Using a constant Courant number for each cell of 20 for the airfoil problem, Table 19 shows the computed sensitivity derivatives of C_L and C_D with respect to T , C , and L , and the number of iterations required by the AF method are also given. As expected, the computed sensitivity derivatives here are in excellent agreement with those reported previously for this problem. Table 20 is a summary of the total CPU times required in this example, comparing the present method with the previously presented results.

Design Variable	Number of Iterations	$\frac{dC_L}{dT, C, L}$	$\frac{dC_D}{dT, C, L}$
T	466	-3.126 E+00	+3.938 E-01
C	428	+3.968 E+00	+3.603 E-01
L	360	-1.816 E-02	-3.290 E-03

Table 19 - NACA 2412 Airfoil Problem, Approximately Factored (AF) Incremental Method, Four OM Error Reduction

Strategy Used	Total CPU Time (Seconds)
Incremental, AF Solver, (4 OM)	50
Incremental, Direct Solver (4 OM)	127
Standard Form, Hybrid Direct/Iterative (4 OM)	191

Table 20 - NACA 2412 Airfoil Problem, Comparison of Total CPU Times

The preceding results are encouraging, and demonstrate the feasibility of the proposed methods. Much work remains in selecting and refining the most efficient algorithms and convergence accelerations methods (such as multigrid, for example) for use in the solution of the aerodynamic sensitivity equations in 2D and 3D.

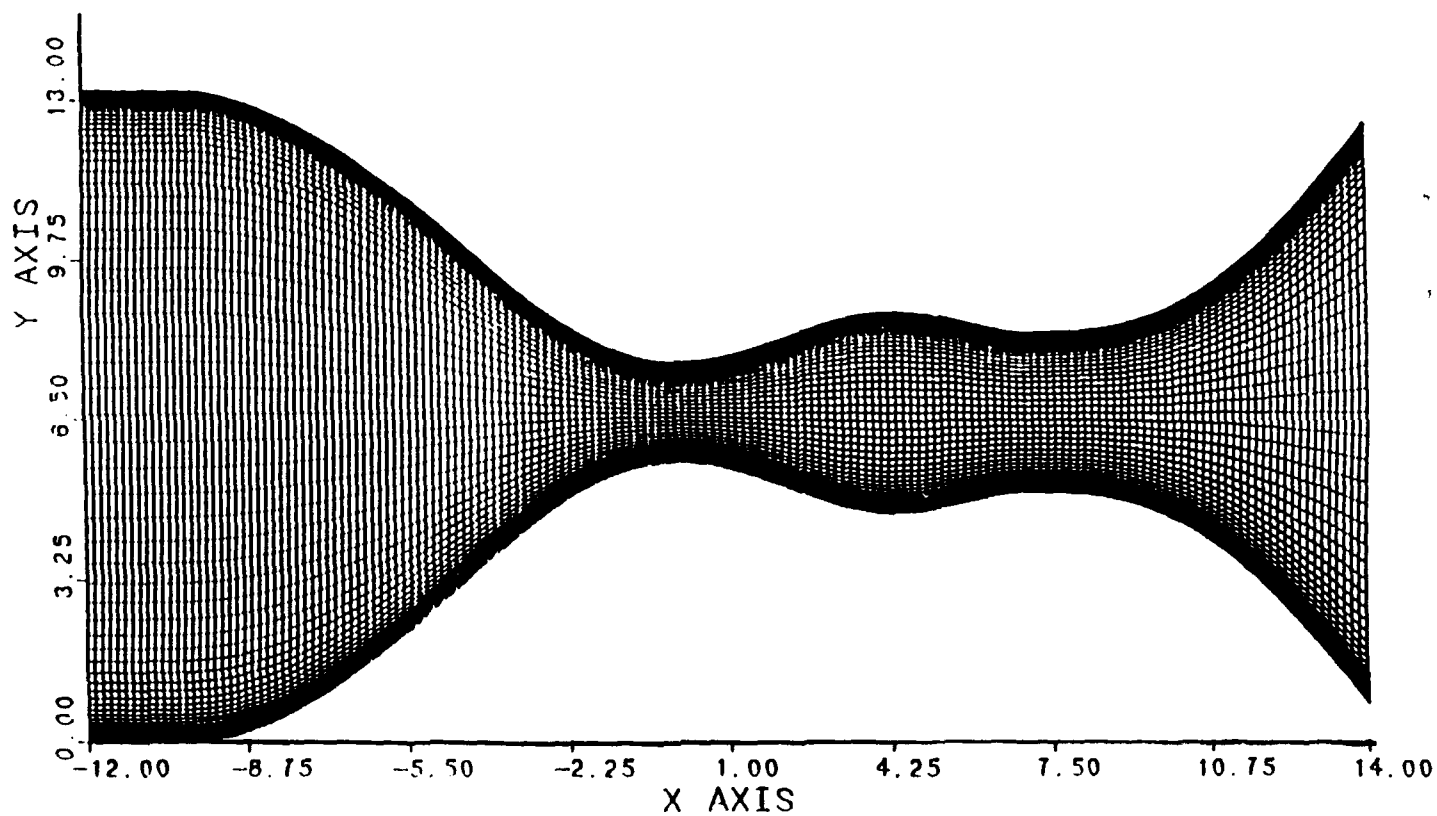


Fig. (1) – Computational Mesh, 171×38 Points,
Geometry of The Double-Throat Nozzle Problem.

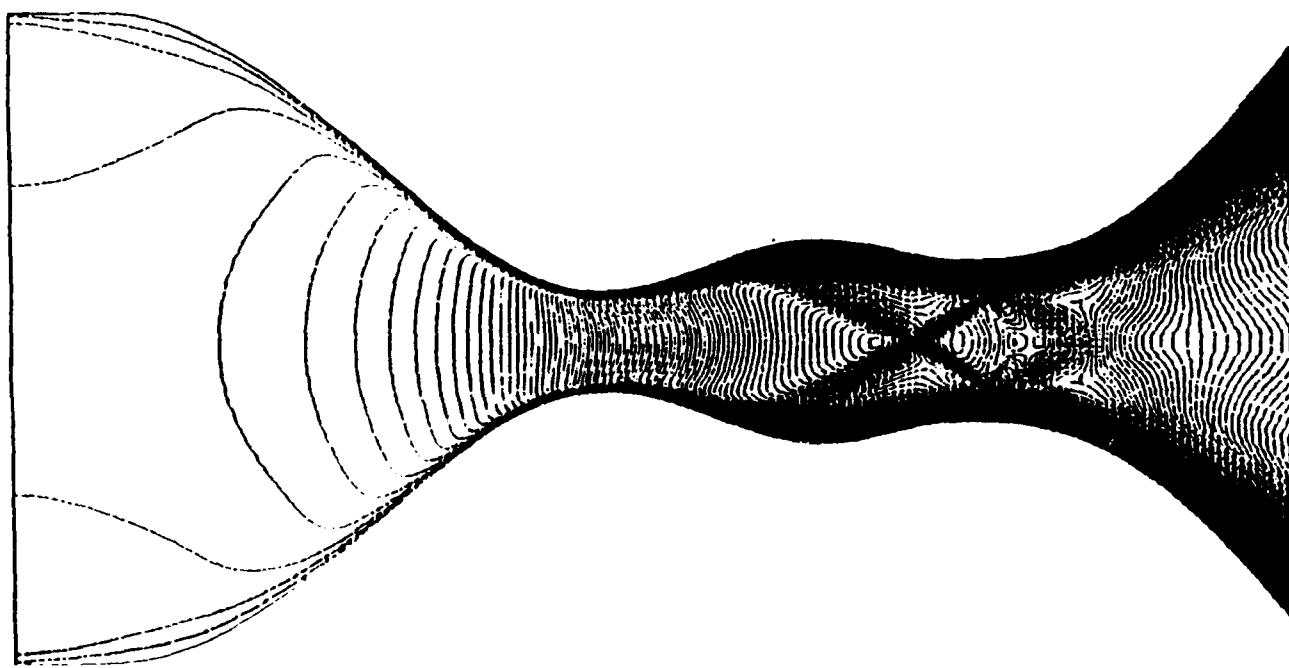


Fig. (2) – Mach Contours For The Double-Throat Nozzle Problem.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE February 1992		3. REPORT TYPE AND DATES COVERED Technical Memorandum	
4. TITLE AND SUBTITLE An Incremental Strategy for Calculating Consistent Discrete CFD Sensitivity Derivatives				5. FUNDING NUMBERS 505-59-53-01	
6. AUTHOR(S) V. M. Korivi; A. C. Taylor III; P. A. Newman; G. J.-W. Hou; and H. E. Jones					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NASA Langley Research Center, Hampton, VA 23665-5225; Aerostructures Directorate, U.S. Army-AVSCOM, Langley Research Center, Hampton, VA 23665-5225				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration, Washington, DC 20546 U.S. Army Aviation Systems Command, St. Louis, MO 63120-1798				10. SPONSORING / MONITORING AGENCY REPORT NUMBER NASA TM-104207 AVSCOM TR-92-B-006	
11. SUPPLEMENTARY NOTES Korivi: ODU, Norfolk, VA; Taylor: ODU, Norfolk, VA; Newman: Langley Research Center, Hampton, VA; Hou: ODU, Norfolk, VA; Jones: Aerostructures Directorate, USA-AVSCOM, Hampton, VA					
12a. DISTRIBUTION / AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category 02				12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) In this preliminary study involving advanced CFD codes, an incremental formulation, also known as the "delta" or "correction" form, is presented for solving the very large sparse systems of linear equations which are associated with aerodynamic sensitivity analysis. For typical problems in 2D, a direct solution method can be applied to these linear equations in either the standard or the incremental form, in which case the two are equivalent. Iterative methods appear to be needed for future 3D applications, however, where direct solver methods require much more computer memory than is currently available. Iterative methods for solving these equations in the standard form result in certain difficulties, such as ill-conditioning of the coefficient matrices, which can be overcome when these equations are cast in the incremental form; these and other benefits are discussed herein. The methodology is successfully implemented and tested in 2D using an upwind, cell-centered, finite volume formulation applied to the thin-layer Navier-Stokes equations. Results are presented for two laminar sample problems: 1) transonic flow through a double-throat nozzle, and 2) flow over an isolated airfoil.					
14. SUBJECT TERMS Sensitivity Analysis Consistent Sensitivity Derivatives				15. NUMBER OF PAGES 33	
				16. PRICE CODE A03	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT		